SYSTEM AND METHOD OF SECURING A COMPUTER FROM UNAUTHORIZED ACCESS

BACKGROUND OF THE INVENTION

1. Field of the invention

The present invention relates generally to computer security and more specifically to making a computer impervious to unwanted users and methods thereof.

2. Description of the Prior Art

In order to maintain a computer server on the Internet, the server generally needs to be secured so that unwanted users will not break into sensitive areas on the server, particularly if the server is being used as an e-commerce server. One way to protect the server is to screen incoming requests with a firewall.

A firewall is a set of related programs, located at a network gateway server that protects the resources of a private network from users from other networks. An enterprise with an intranet that allows its workers access to the wider Internet installs a firewall to prevent outsiders from accessing its own private data resources and for controlling what outside resources its own users have access to.

Basically, a firewall filters all network packets to determine whether to forward them toward their destination. A firewall also includes or works with a proxy server that makes network requests on behalf of workstation users. A firewall is often installed in a specially designated computer separate from the rest of the network so that no incoming request can get directly at private network resources. However, a firewall is generally not impervious to unwanted users.

Since a firewall screens requests, the amount of traffic entering the server slows down considerably. Firewalls can be very complex and expensive, and often require an experienced technician to install and maintain. Furthermore, firewalls are open to attack from hackers, and once penetrated a hacker can gain supervisory rights to the server and access sensitive areas.

Thus, it would be desirable to provide a system and method of securing a computer that does not slow down traffic to the server, is easy to install, easy to use, inexpensive, and impervious to attack by unwanted users.

SUMMARY OF THE INVENTION

The present invention provides a system and method of securing a server computer from unauthorized access without requiring a firewall. The server computer is secured from an external client computer over the Internet or a network by removing the server's root or supervisor rights. The external client computer can be authorized through a trusted IP address list, as well as requiring a password key from the user of the external client computer. A telnet session and an ftp session can remain connected between the server computer and the Internet in order to manage the server computer while it is locked. Even though the supervisor rights have been removed from the server computer, an Internet session will continue to run to allow access to the server computer. The authorized external client can also restore the supervisor rights and manage the web server computer accordingly.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying illustrations. For simplicity and ease of understanding, common numbering of elements is employed where an element is the same in different illustrations.

FIG. 1 is a schematic diagram illustrating a client requesting access to a secure server over the Internet, in accordance with the present invention;

FIG. 2 is a block diagram of the secure server computer shown in FIG. 1, in accordance with the present invention;

FIG. 3 is a block diagram of one embodiment of the non-volatile memory module located within the secure server computer of FIG. 2; and

FIG. 4 is a flowchart of a method illustrating how an administrator can manage and secure the server computer, according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The following is a detailed description of illustrative embodiments of the present invention. As these embodiments of the present invention are described with reference to the aforementioned illustrations, various modifications or adaptations of the methods and or specific structures described may become apparent to those skilled in the art. All such modifications, adaptations, or variations that rely upon the teachings of the present invention, and through which these teachings have advanced the art, are considered to be within the spirit and scope of the present invention. Hence, these descriptions and drawings should not be considered in a limiting sense, as it is understood that the present invention is in no way limited to only the embodiments illustrated.

Referring now to FIG. 1, a schematic diagram illustrates a web server 100 and a client computer 102 connected to the Internet 104. Excellent results can be obtained when the web server 100 is running a Unix® operating system, however, other operating systems such as Windows® can also be used. A qualified user or an administrator using a client computer 102 has the ability to access the server 100 through the Internet 104 in order to manage the server 100 and to pseudo lock the server 100 so that no unauthorized access can be gained.

FIG. 2 is a block diagram of the web server computer 100 shown in FIG. 1. Computer 100 includes a CPU 202, RAM 204, non-volatile memory 206, an input device 208, a display 210, and an Internet interface 212 for providing access to the Internet.

FIG. 3 is a block diagram of one embodiment of the non-volatile memory module 206 located within the web server computer 100 of FIG. 2. The non-volatile memory 206 includes a database of secure keys 302, a listing of trusted IP addresses 304, and an access engine 306. The database of secure keys 302 includes at least one authorized key or password that is known or held by the server administrator. The access engine 306 provides the administrator with various features for managing the web server computer 100, these features include: a remove supervisor rights engine 308, a restore supervisor rights engine 310, and management tools 312.

During the initial installation of the access engine 306 a password or a secure key 302 is established by the server administrator. The access engine 306 is programmed so that it is only accessible from an external client computer having a trusted IP address. The administrator is able to specify IP addresses that would allow access to the access engine 306.

FIG. 4 is a flowchart of a method illustrating how to secure and manage the web server computer from an authorized client computer through the Internet in accordance with the invention. The administrator begins his request for access to the web server

computer from a client computer at step 400 by starting the access engine. Next at step 402 it is determined if the request from the client computer is from a trusted IP address. The web server computer checks to see if the IP address of the requesting client computer is in the list of trusted IP addresses 304.

If the IP address of the requesting client is not in the list of trusted IP addresses 304 then at step 404 the client request to manage the web server computer is rejected. If the IP address of the requesting client is found in the listing of trusted IP addresses 304, then at step 406 a key or password is requested from the client. It is possible for computer hackers to "spoof" an IP address from an untrusted IP address, therefore an additional security measure of requiring a password is provided for a higher level of security.

If the password entered from the client is not in the database of secure keys 302 then at step 404 the client request to manage the web server computer is rejected. If the key entered from the client is in the database of secure keys 302, then the requesting client is authorized to manage the web server computer.

After being authorized to manage the web server computer, at step 410 the administrator decides whether to lock the server. If the administrator decides to lock the server then at step 412 supervisor rights on the web server computer are then physically removed thereby locking the server computer from any unauthorized access, and at step 424 the process ends. Prior to removing the supervisor

rights on the web server, a telnet session and an ftp session are established with the web server so that the web server can still be accessed over the Internet by, and only by, the client 102.

In order to lock the server, the root, or alias root, is physically removed from the server. This requires rewriting the password file without any supervisory rights in it. In a UNIX operating system, in order to physically remove the root or the supervisory rights from the server, the User ID = 0 (UID=0) and the Group ID = 0 (GID=0) are removed from the computer's user list and group list. After the root is removed, the web server computer is functionally dead or secure and no supervisory commands can be issued at the console of the web server, but the telnet session and the ftp session stay connected and allow the trusted client to access the server over the Internet. Even though the server is functionally dead and nobody can access the server as a supervisor, other applications on the web server continue to run and allow access from users on the Internet.

If, at step 410, the administrator does not lock the server, then at step 414 the administrator has the option to unlock the web server if the server has been previously locked. If the administrator chooses to unlock the server then at step 416 supervisor rights on the server are restored, and at step 424 the process ends. In order to restore the supervisor rights, the supervisor is added to the user list and the group list (i.e. UID=0 and GID=0 is added).

If, at step 414, the server is not unlocked, then at step 418 the administrator can choose to process other requests, such as managing the files on the server. At step 420 any requests by the administrator from the trusted client are processed, and at step 424 the process then ends. If no requests are made by the administrator, then at step 422 the access engine goes through error processing and at step 424 the process ends.